

iOS Developer Day at Labitat

Thomas Flummer

Agenda

- This short introduction
- iOS developer hacking
- Lunch at roughly 12:30
- Continued hacking
- 2 short API showcases during the afternoon
- Round up around 17:30

About me...

- My name is **Thomas Flummer**
- I'm doing software development for a living
- Have been using Mac's for quite some time
- Have been using iPhone since 2007
- Signed up for a developer account in 2008
- **But haven't yet managed to get an iOS app finished and out to the masses**

Developing for iOS

- Using Apple Xcode (\$6 or less)
- Primary language is Objective C
- Requires Mac OS X and Intel CPU
- Need payed Developer Account to run app on iOS device (\$99/year)

Alternatives to Xcode

- There are alternative like:
 - Adobe Flash CS5
 - Airplay SDK
 - Dragonfire SDK (remote compile)
 - MonoTouch (C#, though only on Mac)
 - Corona SDK (apps using Lua)
 - and others...
- All the above are commercial solutions

Xcode 4

Newest version of
Apple's IDE



Objective C on iOS

- Objective C is a superset of C
- Syntax is a bit weird in the beginning
- On iOS there is no garbage collector
- Memory Management with reference counting

iOS system architecture

- Cocoa Touch
- Media
- Core Services
- Core OS

Core OS

- Threading (POSIX threads)
- Networking (BSD sockets)
- File-system access
- Standard I/O
- Bonjour and DNS services
- Locale information
- Memory allocation
- Math computations

Core Services

- Core Foundation framework
- Address Book
- Core Location
- CFNetwork
- Security
- SQLite
- XML libraries

Media Layer

- Graphics Technologies for 2D and 3D drawing
 - OpenGL ES 1.1 (& 2.0 on 3GS)
 - Core Animation
 - Quartz
- Core Audio
 - Audio type and file info
 - Playback and Recording
 - Audio Units for Audio processing
- 3D audio in applications
- VideoTechnologies
- Full-screen video playback Playback of movie files: .mov, .mp4, .m4v, and .3gp

Cocoa Touch

- Application management
- Graphics and windowing support
- Event-handling support
- User interface management
- Objects representing the standard system views and controls
- Support for text and web content
- Accelerometer data
- The built-in camera
- The user's photo library
- Device-specific information

The Apple design idea

- Apple has a lot of design guidelines
 - “pick the few features, most frequently used, by the majority of your users, most appropriate for the mobile context”
 - Polish and Refine

Design Pattern

- Model-View-Controller Design Pattern is highly encouraged and often forced at you.
- All the Apple tools and guides are aimed at using the MVC pattern

Demo